The shortest path finding method for MIMD controlled MX Core

Akihiro Kodama* Yuta Mizokami* Mitsutaka Nakano* Masahiro Iida* Toshinori Sueyoshi*
(*Graduate School of Science and Technology, Kumamoto University)

1 Introduction

The massively parallel processor MX Core[1] is developed by Renesas Technology Corp.. This achieves high performance for multimedia processing by the SIMD (Single Instruction Multiple Data) controlled PEs (Processing Elements). We approach the MIMD (Multiple Instruction Multiple Data) controlled MX Core[2] in order to improve the flexibility of data communication between PEs. In this paper, we show a shortest path finding method for MIMD controlled the MX Core.

2 MX Core

Figure 1(a) shows the MX Core overview that has fine-grained 1,024 PEs and data registers. The MX Core uses V-ch in data communication between PEs. These PEs are connected to another PEs of a certain distance by the V-ch, in fig 1(b). The data communication is controlled by the controller. The MX Core communicates with arbitrary distance in single or multi cycle using the V-ch. The MX Core communicates with the same direction and distance in all PEs. By contrast, the data communications of MIMD type MX Core are controlled individually in each PEs. The problem of MIMD architecture is that controlling of data communication is difficult. We should take into account a data collision that some PEs communicates with the same PE simultaneously. For this reason, it takes time $O(n^m)$ to search for an optimum solution for the number of data $m$.

![Figure 1: MX Core architecture](image)

3 Shortest Path Finding Method

In this section, we propose the shortest path finding method for the MIMD controlled MX Core. The shortest path is defined as the minimum communication cycle to destination PE at each PE. Note that we do not take into account the effect of another PEs, that is, the data collision. The critical path is defined as path having the maximum cycle on shortest path in all PEs. The cycle of critical path means the minimum cycle in individual path finding problem at each PE and does not mean always the optimum cycle in the whole path finding problem. However, the communication finished in the cycle of the critical path means an optimal solution.

We apply both a probabilistic decision and a heuristics algorithm to the path finding. First, we search the shortest path in each entry. Second, we set a high priority to longer path. If the data collision occurred, we chose the higher priority path. If it takes time too much to find the path, we perform a division of data. The division of data means dividing the number of the data $m$ into the halves by applying a divide and conquer algorithm. Third, we update the priority concerning selected path. Finally, we repeat this method fifty times updating the priority of path at random to find the better solution, and output the best solution.

4 Evaluation and Result

We performed the path finding for a random data with this method. The random data means a destination of each PE is decided at random. We prepare random data five kinds of the number of data in the following 128, 256, 512, 768 and 1,024. The random data have fifty patterns in each the number of data. We evaluate average of the increasing rate of communication cycle from the cycle of the critical path with this method. We are aimed at reducing the increasing rate.

Table 1 shows the increasing rate for the number of data. In columns 1 and 2 of Table 1, we show the increasing rate. Also, in column 4 of Table 1, we report improvement rate of the best solution from the initial solution with this method.

<table>
<thead>
<tr>
<th>Number of data</th>
<th>Increasing rate(avg.)[%]</th>
<th>Improvement rate[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>2.77</td>
<td>100.00</td>
</tr>
<tr>
<td>256</td>
<td>3.79</td>
<td>91.21</td>
</tr>
<tr>
<td>512</td>
<td>5.71</td>
<td>89.17</td>
</tr>
<tr>
<td>768</td>
<td>39.19</td>
<td>47.33</td>
</tr>
<tr>
<td>1,024</td>
<td>71.00</td>
<td>24.79</td>
</tr>
</tbody>
</table>

We achieve improvement for all data in Table 1. The increasing rate increases with the number of data $m$. This reason is the communication cycle increases by the data division. Increasing the number of data $m$ takes long time to find the solution with this method. For this reason, it is necessary to perform the data division. The communication cycle increases according to the number of division if the data division is performed. Even so, the increasing rate is suppressed to approximately 70% or less with this method.

References